

۱. کامپایل کد منبع:

برنامه‌های اندروید ابتدا به زبان Java نوشته می‌شوند. سپس، کدهای منبع Java به کدهای بایت کد تبدیل می‌شوند.

۲. ساخت DEX فایل: DEX یک فرمت فایل است که برای ذخیره و اجرای کدهای بایت در ماشین دالویک استفاده می‌شود. این فرمت از کدهای بایت Java ساخته می‌شود.

۳. ترجمه به فایل APK:

برنامه به صورت یک فایل (Android Package) APK بسته‌بندی می‌شود. این فایل شامل منابع برنامه، فایل DEX، فایل‌های منابع، تصاویر، و اطلاعات دیگر مورد نیاز برنامه است.

۴. نصب برنامه:

کاربر برنامه را روی یک دستگاه اندرویدی نصب می‌کند. در این مرحله، فایل APK به دستگاه منتقل شده و برنامه روی دستگاه نصب می‌شود.

۵. اجرای برنامه توسط ماشین دالویک:

ماشین دالویک که یک ماشین مجازی است، کدهای بایت DEX را اجرا می‌کند. این ماشین دالویک مسئول اجرای برنامه و ارتباط با سیستم عامل اندروید است.

انتقال از ماشین دالویک به ART در سیستم‌عامل‌های اندروید، بهبودهای مهمی در زمینه امنیت به همراه داشته است. در زیر به چند دلیل امنیتی برای انتقال از دالویک به ART اشاره می‌شود:

۱. کامپایل پیش‌از زمان (AOT): ART از رویکرد کامپایل پیش‌از زمان (AOT) استفاده می‌کند، به این معنا که بایت‌کد در مرحله نصب به کد ماشین نیتیو تبدیل می‌شود و نه در زمان اجرا. این امر باعث کاهش هزینه پردازش در زمان اجرا می‌شود و باز کردن فرصت کاهشی را برای حملات یا تغییرات در طول اجرا مشکل‌تر می‌کند.

۲. بهبود عملکرد برنامه:

استفاده از رویکرد AOT توسط ART، باعث بهبود عملکرد برنامه نسبت به روش Just-In-Time (JIT) استفاده شده توسط دالویک می‌شود. اجرای سریع، پنجره فرصت برخی از نوع حملات را کاهش می‌دهد و به افزایش امنیت سیستم کمک می‌کند.

۳. بهبود مکانیزم جمع‌آوری زباله: ART مکانیزم جمع‌آوری زباله موثرتری دارد که در مدیریت حافظه بهبود می‌آورد. این امر می‌تواند خطرات مربوط به حافظه مانند نشت حافظه یا سرریز بوفر را کاهش دهد و به محیط اجرایی ایمن‌تری کمک کند.

۴. تصادفی‌سازی آدرس فضای لیه (ASLR): ART امنیت سیستم اندروید را با پیاده‌سازی تصادفی‌سازی آدرس فضای لیه (ASLR) بهبود می‌بخشد. ASLR ترتیب حافظه فرآیندها را تصادفی می‌کند که برای حمله‌کنندگان دشوارتر می‌شود تا مکان خاصی از کد یا داده را پیش‌بینی کنند و در نتیجه، خطر بهره‌برداری موفق از آسیب‌پذیری‌ها را کاهش می‌دهد.

۵. بهبود ویژگی‌های امنیتی در نسخه‌های اندروید: ART با ویژگی‌های امنیتی افزوده شده در نسخه‌های جدیدتر اندروید یکپارچه شده است. انتقال به ART اطمینان می‌دهد که برنامه‌ها می‌توانند از آخرین ویژگی‌ها و بهبودهای امنیتی ارائه شده توسط پلتفرم اندروید بهره‌مند شوند.

۶. تطابق با کتابخانه‌ها و ویژگی‌های امنیتی: ART به صورت هماهنگ‌تر با کتابخانه‌ها و ویژگی‌های امنیتی طراحی شده است، که به توسعه‌دهندگان این امکان را می‌دهد که از آخرین API‌ها و ابزارهای امنیتی استفاده کنند و امنیت برنامه‌های خود را تقویت کنند.

۷. جداسازی و سندباکسینگ: ART بهبودهایی در جداسازی و سندباکسینگ برنامه‌ها دارد، که امنیت برنامه‌های فردی را افزایش می‌دهد. این به کاهش دسترسی یا اختلاط با داده‌ها و منابع برنامه‌های دیگر در دستگاه کمک می‌کند.

کلیتاً، انتقال از دالویک به ART نه تنها بهبودهای عملکردی به همراه دارد بلکه به بهبودهای امنیتی نیز می‌انجامد که به ایجاد یک محیط اجرایی اندروید محکم‌تر و امن‌تر کمک می‌کند.

دو زبان XML و HTML متفاوت هستند که برای اهداف مختلف در دنیای وب و مدیریت داده‌ها استفاده می‌شوند. (XML (Extensible Markup Language) به عنوان یک زبان علامت‌گذاری قابل توسعه طراحی شده است و هدف اصلی آن انتقال و ذخیره داده‌ها با ساختاری کاملاً قابل تعریف و انعطاف‌پذیر است. XML تمرکز خود را بر روی تبادل داده‌ها و توصیف ساختار آنها قرار داده و هیچ فرضیه گرافیکی یا نمایشی ندارد.

به عنوان مقابل، HTML (HyperText Markup Language) برای ساختاردهی و نمایش اطلاعات در صفحات وب به کار می‌رود. HTML به وسیله تگ‌ها و ویژگی‌های گرافیکی مانند لینک‌ها، تصاویر و فرم‌ها، اطلاعات را به صورت مناسب برای مرورگرها ساختاردهی می‌کند. در واقع، HTML یک زبان نشانه‌گذاری فرعی است که مختص نمایش اطلاعات بر روی صفحات وب می‌باشد و به کمک تگ‌ها و خصوصیات گرافیکی، صفحات وب را شکل می‌دهد.

در محیط سیستم‌عامل، دو مفهوم مهم به نام **درخواست** و **فرمان** وجود دارند که هرکدام در روند اجرای عملیات‌ها نقش مهمی ایفا می‌کنند.

درخواست، نمایانگر یک تقاضا یا درخواست از سوی برنامه‌ها یا کاربران به سیستم‌عامل است تا یک **خدمت یا عملیات خاص اجرا شود**. به عبارت دیگر، یک فرآیند یا کاربر از سیستم خواهان اجرای یک عمل معین می‌شود و این درخواست به سیستم‌عامل ارسال می‌شود. مثلاً، یک برنامه ممکن است از سیستم درخواست بدهد تا یک فایل را باز کند یا از یک منبع خاص بهره‌مند شود.

از سوی دیگر، **فرمان** یک دستور **مستقیم** است که توسط **کاربر** یا **برنامه** برای سیستم‌عامل ارسال می‌شود. این دستورات به سیستم می‌گویند که **یک عمل خاص را انجام دهد**. به عنوان مثال، دستورات مانند "copy" برای کپی کردن فایل، "move" برای جابجایی فایل‌ها، یا "delete" برای حذف فایل‌ها از این دستورات هستند.

درخواست برای جستجوی داده‌ها استفاده می‌شود. به عنوان مثال، در شبکه، درخواست HTTP برای گرفتن داده از سرور استفاده می‌شود. اما یک فرمان به سیستم داده می‌شود تا یک وظیفه خاص را انجام دهد، مانند دستورهای که در ترمینال لینوکس یا خط فرمان ویندوز استفاده می‌شود. درخواست‌ها می‌توانند شامل چندین فرمان باشند. به طور مثال، یک درخواست واکنش معمولاً به عنوان یک درخواست به یک واکنش تصویر می‌شود، اما قبل از بازگشت نهایی به کلاینت، دستورهای مشابهی بین مشتری و سرور تبادل می‌شوند.

در کل، **درخواست** به عنوان **یک نیاز به خدمات سیستمی** دیده می‌شود که از طریق یک درخواست مشخص معلوم می‌شود.

در مقابل، **فرمان** به عنوان **یک دستور روشن و صریح** برای انجام یک وظیفه یا عملیات خاص در سیستم‌عامل در نظر گرفته می‌شود.